MCP23017 DFROBOT

PCP23017 DFROBOT EN

introduction

Il s'agit d'une carte d'extension IO basée sur MCP23017 qui fournit 16 ports IO supplémentaires pour votre microcontrôleur. Il peut être configuré pour 8 adresses différentes, ce qui signifie que 8 modules peuvent être connectés via deux bus IIC pour atteindre jusqu'à 128 extensions IO. Ce module est une excellente solution au problème d'insuffisance d'E/S dans des applications telles que les robots et les médias interactifs.

Cette carte d'extension IO dispose de 2 groupes de broches IO : GPIOA et GPIOB. Chaque groupe dispose de 8 E/S indépendantes et chacune d'entre elles peut être arbitrairement configurée comme entrée ou sortie, et une entrée pull-up (connectée à 100KΩ pull-up), interruption et ainsi de suite. En plus de cela, le module possède 2 broches de signal d'interruption IA et IB qui sont utilisées pour détecter l'interruption dans les ports GPIOA et CPIOB respectivement. Lorsqu'une interruption se produit sur l'une des broches de CPIOA ou GPIOB, la broche IA ou IB produira un signal de niveau haut en conséquence. Applications

- Robots
- Média interactif
- Cube lumineux

spécification

- Alimentation : 3,3 V-5,5 V
- Adresse IIC : 0x20~0x27
- E/S numériques/Entrée/Sortie : PA0-PA7, PB0-PB7

- Courant de conduite IO : 20 mA
- Broche d'interruption de signal : IA, IB
- Dimensions : 44x32mm / 1.73×1.26“

Présentation du conseil

Numéro	Nom	La description
1	+/CCV	Positif
2	-/GND	Négatif
3	C	Ligne d'horloge IIC
4	D	Ligne de données IIC
5	RST	Broche de réinitialisation
6	IA	Broche de détection d'interruption GPIOA
7	IB	Broche de détection d'interruption GPIOB
8	REP	Indicateur d'alimentation
9	A0/A1/A2	Commutateur d'adresse IIC

Tableau des ports E/S numériques

CPIOA	**0**	**1**	**2**	**3**	**4**	**5**	**6**
	PA0	PA1	PA2	PA3	PA4	PA5	PA6
OPCIP	**8**	**9**	**dix**	**11**	**12**	**13**	**14**
	PB0	PB1	PB2	PB3	PB4	PB5	PB6

Chaque broche numérique de la table peut être arbitrairement configurée comme entrée ou sortie, et comme entrée pull-up, interruption, etc.

Didacticiel

Conditions

- Matériel
  - DFRduino UNO R3 (ou similaire) x 1
  - MCP23017 Module d'extension IIC à 16 E/S numériques x1
  - Boutons
  - Modules DEL
- Logiciel
  - EDI Arduino
  - Téléchargez et installez la bibliothèque MCP23017 (A propos de comment installer la bibliothèque ?)

Liste des fonctions API

fonctionsAPI.c

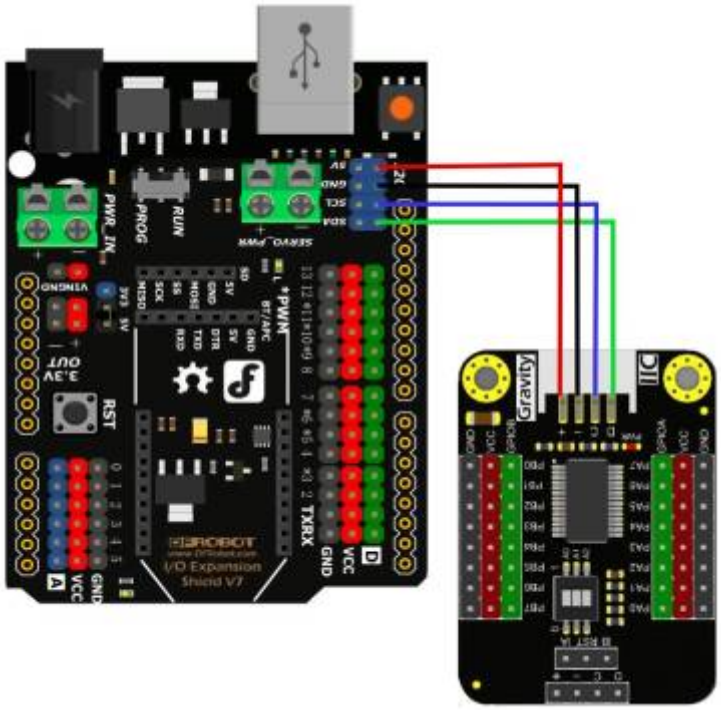
```
 /**
   * @brief Set the pin mode to  input, output or pull-up input
(internal 100KΩ pull-up resistor)
   * @param pin Pin number, it could be all enumeration values (eGPA0-
eGPB7/ 0-15) included in ePin_t.
   * @param mode Mode, it can be set to Input, Output, Pull-up Input
(internal 100KΩ pull-up resistor)
   * @return Return 0 if the setting is successful, otherwise return
non-zero.
   */
  int pinMode(ePin_t pin, uint8_t mode);

  /**
   * @brief Write digtial pin. The pin needs to be set to output mode
before writing.
   * @param pin Pin number, it could be all enumeration values (eGPA0-
eGPB7/ 0-15) inlcuded in ePin_t.
   * @param level High level 1 or Low level 0
   * @return Return 0 if the writing is successful, otherwise return
non-zero.
   */
  int digitalWrite(ePin_t pin, uint8_t level);

  /**
   * @brief Read digital pin. The pin needs to be set to input mode
before reading.
   * @param pin Pin number, it could be all enumeration values (eGPA0-
eGPB7/ 0-15) included in ePin_t.
   * @return Return High or Low
   */
  int digitalRead(ePin_t pin);

  /**
   * @brief Set a pin to interrupt mode
   * @param pin Pin number, it could be all enumeration values (eGPA0-
eGPB7/ 0-15) included in ePin_t.
   * @param mode Interrupt mode: all enumeration values included in
eInterruptMode_t.
   * @param cb Interrupt service function, needs to be defined and
transferred parameter by users. Prototype: void func(int)
   */
  void pinModeInterrupt(ePin_t pin, eInterruptMode_t mode,
MCP23017_INT_CB cb);

  /**
   * @brief Poll if an interrupt occurs on a port group.
   * @param group Port group, it could be all enumeration values
included in eGPIOGroup_t,  GPIO GroupA(eGPIOA),
   * @n GPIO GroupB(eGPIOB) GroupA+B (eGPIOALL).
   * @n When setting to eGPIOA□poll if an interrupt occurs on the port
group A.
```

```
   * @n When setting to eGPIOB, poll if an interrupt occurs on the port
group B.
   * @n When setting to eGPIOALL, poll if an interrupt occurs on the
port group A+B
   * @n None, poll if an interrupt occurs on the all ports of group A
and B by default.
   */
  void pollInterrupts(eGPIOGroup_t group=eGPIOALL);

  /**
   * @brief Convert pin into string description
   * @param pin Pin number, it could be all enumeration values (eGPA0-
eGPB7/ 0-15) inlcuded in ePin_t.
   * @return Return pin description string
   * @n such as "eGPA0" "eGPA1" "eGPA2" "eGPA3" "eGPA4" "eGPA5" "eGPA6"
"eGPA7"
   * @n   "eGPB0" "eGPB1" "eGPB2" "eGPB3" "eGPB4" "eGPB5" "eGPB6"
"eGPB7"
   * @n   "eGPA" "eGPB"
   */
  String pinDescription(ePin_t pin);

  /**
   * @brief Convert pin into string description
   * @param pin  Pin number, range 0~15
   * @return Return pin description string
   * @n such as "eGPA0" "eGPA1" "eGPA2" "eGPA3" "eGPA4" "eGPA5" "eGPA6"
"eGPA7"
   * @n   "eGPB0" "eGPB1" "eGPB2" "eGPB3" "eGPB4" "eGPB5" "eGPB6"
"eGPB7"
   * @n   "eGPA" "eGPB"
   */
  String pinDescription(int pin);
```

La relation entre l'adresse DIP et IIC

A2	A1	A0	Adresse IIC
0	0	0	0x20
0	0	1	0x21
0	1	0	0x22
0	1	1	0x23
1	0	0	0x24
1	0	1	0x25
1	1	0	0x26
1	1	1	0x27 (par défaut)

Diagramme de connexion



Exemple de code 1 - Entrée du bouton

Réglez la broche PA0 de la carte en mode entrée, connectez-la avec un bouton. Lorsque le bouton est enfoncé, la série imprime la chaîne “Button press!”.

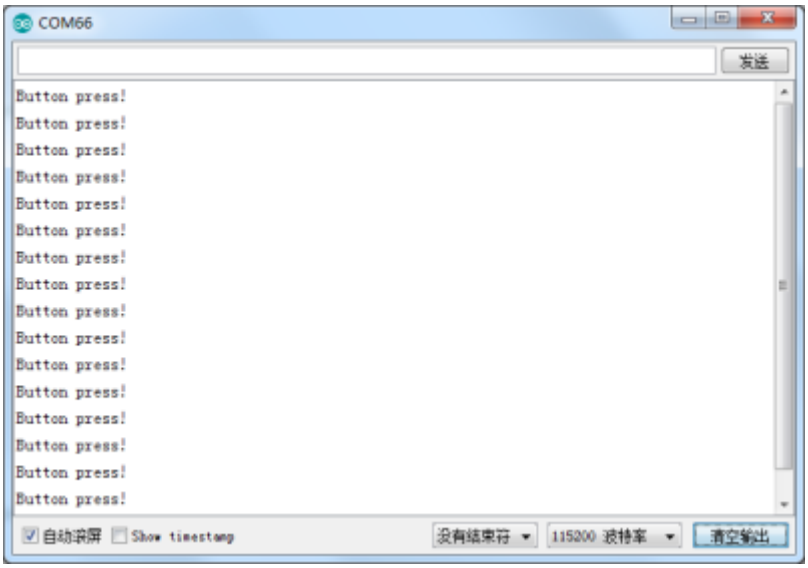
Bouton1.ino

```
/*!
 * @file buttonInput.ino
 * @brief Connect a button to the IO expansion board, set a pin of the
board(eg: eGPA0) to input mode to detect the button status.
 * @n Experiment phenomenon: connect a button on a pin of the IO
board(eg:eGPA0), detect the level of the pin and print out
 * @n the button status on serial port.
 *
 * @copyright   Copyright (c) 2010 DFRobot Co.Ltd
(https://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [Arya](xue.peng@dfrobot.com)
 * @version  V1.0
 * @date  2019-07-18
 * @get from https://www.dfrobot.com
 * @url https://github.com/DFRobot/DFRobot_MCP23017
 */
#include <DFRobot_MCP23017.h>
/*DFRobot_MCP23017 constructor
 *Parameter&wire Wire
```

```
 *Parameter addr  I2C address can be select from 0x20~0x27; the
relationship of DIP switch(A2, A1, A0) and I2C address(0x27) is
 *shown below:
  * 0  0  1  0  | 0  A2 A1 A0
    0  0  1  0  | 0  1  1  1    0x27
    0  0  1  0  | 0  1  1  0    0x26
    0  0  1  0  | 0  1  0  1    0x25
    0  0  1  0  | 0  1  0  0    0x24
    0  0  1  0  | 0  0  1  1    0x23
    0  0  1  0  | 0  0  1  0    0x22
    0  0  1  0  | 0  0  0  1    0x21
    0  0  1  0  | 0  0  0  0    0x20
 */
DFRobot_MCP23017 mcp(Wire, /*addr =*/0x27);//constructor, change the
Level of A2, A1, A0 via DIP switch to revise the I2C address within
0x20~0x27.
//DFRobot_MCP23017 mcp;//use default parameter, Wire  0x27(default I2C
address)

//Prepare: connect a button to a digital pin of the IO expansion
board(eg: eGPA0)
void setup() {
  Serial.begin(115200);
  /*wait for the chip to be initialized completely, and then exit*/
  while(mcp.begin() != 0){
    Serial.println("Initialization of the chip failed, please confirm
that the chip connection is correct!");
    delay(1000);
  }
  /*pinMode function is used to set the pin mode of module
  Parameter pin, the available parameter is shown below:
  eGPA0  eGPA1  eGPA2  eGPA3  eGPA4  eGPA5  eGPA6  eGPA7  eGPA
   0       1      2      3      4      5      6      7
  eGPB0  eGPB1  eGPB2  eGPB3  eGPB4  eGPB5  eGPB6  eGPB7  eGPB
   8       9      10     11     12     13     14     15
  Parameter mode, can be set to: INPUT, OUTPUT, INPUT_PULLUP(internal
100KΩ pull-up resistor)
  */
  mcp.pinMode(/*pin = */mcp.eGPA0, /*mode = */INPUT);
  /*Set all Group GPIOA pins to input*/
  //mcp.pinMode(/*pin = */mcp.eGPA, /*mode = */INPUT);
}

void loop() {
  /*digitalRead function is used to read the Level of a digital pin.
The pin needs to be set to input mode before using this function.
  Parameter pin, the available parameter is shown below:
  eGPA0  eGPA1  eGPA2  eGPA3  eGPA4  eGPA5  eGPA6  eGPA7  eGPA
   0       1      2      3      4      5      6      7
```

```
  eGPB0  eGPB1  eGPB2  eGPB3  eGPB4  eGPB5  eGPB6  eGPB7  eGPB
   8       9      10     11     12     13     14     15
  */
  uint8_t value = mcp.digitalRead(/*pin = */mcp.eGPA0);
  /*Read  level of Group GPIOA pins*/
  //value = mcp.digitalRead(/*pin = */mcp.eGPA);
  if(value){
      Serial.println("Button press!");
      delay(200);
  }else{
      //Serial.println("Button release!");
  }
}
```

- **Résultats attendus** Lorsque le bouton est enfoncé, imprimez le “Button press” sur le port série.



Exemple de code de sortie à 2 broches

Réglez la broche PA7 de la carte en mode sortie, connectez-la avec une LED et changez l'état de la LED en une minute.

2broches.ino

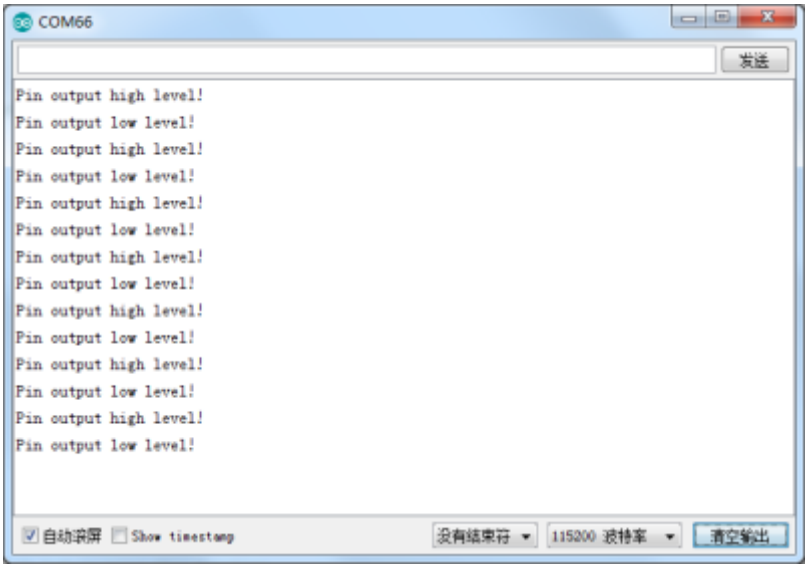
```
/*!
 * @file ledOutput.ino
 * @brief Set a pin of IO expansion board(eg:eGPA7) to output mode, and
output High/Low.
 * @n Experiment phenomenon: the LED connected to the pin of IO
board(eg:eGPA7) repeatedly lights up for 1s and turns off 1s.
 *
 * @copyright   Copyright (c) 2010 DFRobot Co.Ltd
(https://www.dfrobot.com)
```

```
 * @licence     The MIT License (MIT)
 * @author [Arya](xue.peng@dfrobot.com)
 * @version  V1.0
 * @eGPAte  2019-07-18
 * @get from https://www.dfrobot.com
 * @url https://github.com/DFRobot/DFRobot_MCP23017
 */

#include <DFRobot_MCP23017.h>
/*DFRobot_MCP23017 Constructor
 *Parameter &wire  Wire
 *Parameter addr  I2C address can be selected from 0x20~0x27, the
relationship of DIP switch (A2,A1,A0) and I2C address (0x27)is shown
below
  * 0  0  1  0  | 0  A2 A1 A0
    0  0  1  0  | 0  1  1  1    0x27
    0  0  1  0  | 0  1  1  0    0x26
    0  0  1  0  | 0  1  0  1    0x25
    0  0  1  0  | 0  1  0  0    0x24
    0  0  1  0  | 0  0  1  1    0x23
    0  0  1  0  | 0  0  1  0    0x22
    0  0  1  0  | 0  0  0  1    0x21
    0  0  1  0  | 0  0  0  0    0x20
 */
DFRobot_MCP23017 mcp(Wire, /*addr =*/0x27);//constructor, change the
Level of A2, A1, A0 via DIP switch to revise I2C address within
0x20~0x27
//DFRobot_MCP23017 mcp;//use default parameter, Wire  0x27(default I2C
address)

//Prepare: connect the LED to a digital pin of IO expansion
board(eg:eGPA7)
void setup(void)
{
  Serial.begin(115200);
  /*wait for the chip to be initialized completely, and then exit*/
  while(mcp.begin() != 0){
    Serial.println("Initialization of the chip failed, please confirm
that the chip connection is correct!");
    delay(1000);
  }
  /*pinMode function is used to set the pin mode of the module
  Parameter pin, the available parameter is shown below:
  eGPA0  eGPA1  eGPA2  eGPA3  eGPA4  eGPA5  eGPA6  eGPA7  eGPA
   0       1      2      3      4      5      6      7
  eGPB0  eGPB1  eGPB2  eGPB3  eGPB4  eGPB5  eGPB6  eGPB7  eGPB
   8       9      10     11     12     13     14     15
  Parameter mode, can be set to: INPUT, OUTPUT, INPUT_PULLUP mode
(internal 100KΩ pull-up resistor)
```

```
  */
  mcp.pinMode(/*pin = */mcp.eGPA7, /*mode = */OUTPUT);
  /*Set all Group GPIOA pins to output*/
  //mcp.pinMode(/*pin = */mcp.eGPA, /*mode = */OUTPUT);
}

void loop(void)
{
  Serial.println("Pin output high level!");
  /*digitalWrite function is used to make the pin output HIGH or LOW.
The pin needs to be set to output mode before using this function.
  Designate a pin on the IO expansion board; parameter pin, the
available parameter is shown below:
  eGPA0  eGPA1  eGPA2  eGPA3  eGPA4  eGPA5  eGPA6  eGPA7  eGPA
   0      1      2      3      4      5      6      7
  eGPB0  eGPB1  eGPB2  eGPB3  eGPB4  eGPB5  eGPB6  eGPB7  eGPB
   8      9      10     11     12     13     14     15
  */
  mcp.digitalWrite(/*pin = */mcp.eGPA7, /*level = */HIGH);
  /*Set GPIOIA0-GPIOIA3 to low and GPIOIA4-GPIOIA7 to high*/
  //mcp.digitalWrite(/*pin = */mcp.eGPA, /*Port Value = */0xF0);
  delay(1000);
  Serial.println("Pin output low level!");
  mcp.digitalWrite(/*pin = */mcp.eGPA7, /*level = */LOW);
  /*Set GPIOIA0-GPIOIA3 to high and GPIOIA4-GPIOIA7 to low*/
  //mcp.digitalWrite(/*pin = */mcp.eGPA, /*Port Value = */0x0F);
  delay(1000);
}
```

- **Résultats attendus**

La LED connectée à la broche PA7 change d'état en une minute, imprime en série la sortie de niveau de la broche PA7.



Exemple de code - interrogation d'interruption

Réglez les broches IO de la carte en mode interruption, comme eChangeLevel, eFalling, eRising, eHighLevel, eLowLevel. Réglez PA0, PA1, PB6, PB7 sur eChangeLevel, eFalling, eRising, eHighLevel respectivement, interrogez et imprimez en série les broches où l'interruption s'est produite.

interruption1.ino

```
/*!
 * @file pollInterrupt.ino
 * @brief Set a pin of IO expansion board to interrupt mode, poll if
there is an interrupt occurring on the pins of the port.
 * @n Experiment phenomenon: poll if there is an interrupt occurring on
the port group(A, B, A+B), if there is, serial print the pin which is
interrupted.
 *
 * @copyright   Copyright (c) 2010 DFRobot Co.Ltd
(https://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [Arya](xue.peng@dfrobot.com)
 * @version  V1.0
 * @date  2019-07-18
 * @get from https://www.dfrobot.com
 * @url https://github.com/DFRobot/DFRobot_MCP23017
 */
#include <DFRobot_MCP23017.h>
DFRobot_MCP23017 mcp(Wire, 0x27);//constructor, change the Level of A2,
A1, A0 via DIP switch to revise I2C address within 0x20~0x27.
//DFRobot_MCP23017 mcp;//use default parameter, Wire  0x27(default I2C
address)

/*Interrupt service function, prototype void func(int index), index:
the number of the pin that is interrupted*/
void func(int index){
  String description = mcp.pinDescription(/*pin = */index);
  Serial.print(description);
  Serial.println(" Interruption occurs!");
}

void setup() {
  Serial.begin(115200);

  /*wait for the chip to be initialized completely, and then exit*/
  while(mcp.begin() != 0){
    Serial.println("Initialization of the chip failed, please confirm
that the chip connection is correct!");
    delay(1000);
  }
```

```
  /*Parameter mode, the available is shown below:
  eLowLevel             eHighLevel             eRising
eFalling                eChangeLevel
  Low-level interrupt   High-level interrupt   Rising-edge interrupt
Falling-edge interrupt   Double-edge interrupts
  Parameter cb Interrupt service function(with parameter)
  Prototype void func(int)
  */
  mcp.pinModeInterrupt(/*p = */mcp.eGPA0, /*mode = */mcp.eChangeLevel,
/*cb = */func);//digital pin0(eGPA0), double edge interrupt, generate
an interrupt when the status of pin0 changes, INTA output High level.
  mcp.pinModeInterrupt(/*p = */mcp.eGPA1, /*mode = */mcp.eFalling, /*cb
= */func);//digital pin1(eGPA1), falling edge interrupt, generate an
interrupt when the status of pin 1 changes from High to Low, INTA
output High level.
  mcp.pinModeInterrupt(/*p = */mcp.eGPB7, /*mode = */mcp.eRising, /*cb
= */func);//digital pin15(eGPB7), rising edge interrupt, generate an
interrupt when the status of pin15 changes from Low to High, INTB
output High level.
  mcp.pinModeInterrupt(/*p = */mcp.eGPB6, /*mode = */mcp.eHighLevel,
/*cb = */func);//digital pin14(eGPB6), high level interrupt, generate
an interrupt when the pin 14 is in high level, INTB output High level.
}

void loop() {
  /*pollInterrupts function is used to poll if an interrupt occurs on a
port group
  Parameter group, the available parameter is shown below: (default:
eGPIOALL)
   eGPIOA          eGPIOB              eGPIOALL
   Port Group A    Port Group B        Port Group A+B
  */
  mcp.pollInterrupts();
 // delay(1000);
}
```